Encoding Relation Requirements for Relation Extraction: A Semantic Loss Perspective

Anonymous EMNLP submission

Abstract

Identifying relations between entities is crutial for knowledge base population, question answering and etc. Most existing relation extractors make predictions purely based on the input natual language text, and ignores the argument type and cardinality constraints required by each relation. In this paper, we propose to format these constraints into boolean expressions, and incorporating them to existing relation extractors by introducing a new term to the original loss function. Our method do not introduces extra cost to the prediction phase and is plug-and-play for most existing relation extraction models. Experiments on both English and Chinese datasets show that our method improves the base model by a clear margin.

1 Introduction

Relation extraction (RE) aims at identifying relations between pairs of entities from raw text, and its success can benefit many knowledge base (KB) related tasks like KB population, question answering (QA) and etc (Suchanek et al., 2013).

In the literature, RE is usually investigated in the distant supervision (DS) paradigm, where datasets are automatically constructed by aligning existing KB *(subject, relation, object)* triples with a large corpus, and considers sentences containing the subject and object entity in a triple as evidence for the corresponding relation (Riedel et al., 2010). To alleviate the sentence level noise in the automatically constructed dataset, RE is often considered in the multi-instance learning (MIL) framework where all the sentences containing the target subject and object are packed into a ***sentence bag***, and relation extractors take in these sentences to predict the relations for the entity pair. Following this framework, Zeng et al. (2015) uses piecewise convolutional neural network (PCNN) to handle the extraction task, Lin et al. (2016) introduces attention mechanism for better noise tolerence, Ye et al. (2017) makes further improvements by learning co-occurence tendency between relations via learning to rank.

Typically, most existing relation extractors relies only on input sentences to make predictions, and ignores the constraints required by each relation. Take the relation *Capital* for example, it would expect its subject to be a country and its object to be a city. And in most cases, it also expects a city to be the capital of only one country. This kind of constraints can help us identify inconsistent predictions and thereby improve the extraction results.

However, properly utilizing these clues is nontrivial, since many KBs do not have a well-defined typing system or a cardinality specification for relations. Chen et al (2014) evades this difficulty by implicitly mining such requirements from data. Specifically, while collecting cardinality requirements directly from data, they evades the tricky argument type constraints by collecting relations pairs that can not share the same subject or object instead, which implicitly captures the argument type requirement. After that, they collects all the extraction results, and uses integer linear programming (ILP) to resolve the inconsistencies. However, since ILP operates on the post-processing phase, their method typically requires more time during prediction, and introduces high delay to the use of the extraction results since we need to wait for all the extractions to complete before the ILP step.

To overcome the problems of the ILP method, we propose to incorporate these constraints by introducing an additional loss during training, and the test phase thus incurs no extra costs. Specifically, we format argument type and cardinality constraints into propositional logic sentences, and

use the semantic loss framework (Xu et al., 2017) to convert the constraints into a loss term, which penalizes inconsistent predictions during training. In this way, the classification boundary is made more discriminative and therefore lead to better generalization ability of the model. Since we only add a loss term to the base model, our method is plug-and-play for most relation extraction models. We conduct experiments on both English and Chinese datasets, and the experimental results show that our method can clearly improve the base model, and delivers superior performance compared to the ILP method.

2 Our Approach

In this section, we describe a novel loss term that incorporates the argument type and cardinality constraints required by each relation, which significantly increases the quality of relation extraction results with only moderate extra cost during training. Before delving into the details of our method (Sec. 2.3), we will first briefly introduce our base relation extraction model (Sec. 2.1), and the constraints that we will use as well (Sec. 2.2).

2.1 Base Model

Although our loss term is compatible to most existing relation extractors, in this paper, we take the attentive piecewise convolutional neural network (APCNN) (Lin et al., 2016) as our base model, which is currently one of the most widely used extractor in the relation extraction task.

Specifically, APCNN operates in the MIL framework that takes in all the sentences mentioning the target subject and object, and output the relations between these two entities. We first use the piecewise convolutional neural network (PCNN) (Zeng et al., 2015) to obtain the embedding of each sentence. Then, an attention layer is applied to these sentence embeddings to selectively aggregate them into a sentence bag embedding, which is then fed to a softmax classifier to generate the predicted relation distribution, $\boldsymbol{p}$.

2.2 Relation Constraints

Following Chen et al. (2014), we derive type and cardinality constraints from an existing KB to implicitly capture the expected type and cardinality requirements for the arguments of a relation.

Type Constraint Similar to Chen et al. (2014), we also use entity sharing between relations to implicitly capture the expected argument type of each relation. Specifically, if the subject set of relation R_1 in KB has a large intersection with those in R_2, then we consider R_1 and R_2 have the same expected subject type. We thereby collect relation paris that have the same subject type ($\boldsymbol{C}^{ts}$), object type ($\boldsymbol{C}^{to}$), and relation pairs whose subject type of R_1 is the same as the object type of R_2 ($\boldsymbol{C}^{tso}$). Then, $\boldsymbol{C}^{ts}$, $\boldsymbol{C}^{to}$ and $\boldsymbol{C}^{tso}$ represents the constraints that we expect (r_1, r_2) of the predicted $(subj_1, r_1, obj_1)$, $(subj_2, r_2, obj_2)$ pairs fall into $\boldsymbol{C}^{ts}$ when $subj_1 = subj_2$, $\boldsymbol{C}^{to}$ when $obj_1 = obj_2$ and $\boldsymbol{C}^{tso}$ when when $subj_1 = obj_2$ respectively. The idea is that, if the predicted relations of two triples require the same entity to belong to different types, then at least one of the prediction must be wrong.

Cardinality Constraint Given a subject (or object), some relations should only have one object (or subject). For example, the relation *Capital* would expect only one object for a given subject. Following this observation, we collect relations that can have multiple objects ($\boldsymbol{C}^{co}$) or subjects ($\boldsymbol{C}^{cs}$), and $\boldsymbol{C}^{co}$ (or $\boldsymbol{C}^{cs}$) represents the constraint that we expect the relation of the predicted $(subj_1, r, obj_1)$, $(subj_2, r, obj_2)$ pairs fall into $\boldsymbol{C}^{co}$ (or $\boldsymbol{C}^{cs}$) when $subj_1 = subj_2 \wedge obj_1 \neq obj_2$ (or $subj_1 \neq subj_2 \wedge obj_1 = obj_2$)

2.3 Incorporating Constraints for Training

In this section, we demonstrate our method of converting the relation constraints into a loss term using the semantic loss framework (Xu et al., 2017). We also make suitable modifications to speed up the training process with only minor drop in prediction ability.

Relation Constraint Loss Semantic loss is a general framework that can encode a propositional logic constraints as a loss term in a principled way. Concretely, the semantic loss $L^s(\boldsymbol{C}, \boldsymbol{p})$ is defined as:

$$L^s(\boldsymbol{C}, \boldsymbol{p}) = -log \sum_{\boldsymbol{x} \models \boldsymbol{C}} \prod_{i:\boldsymbol{x} \models X_i} p_i \prod_{i:\boldsymbol{x} \models \neg X_i} (1 - p_i) \quad (1)$$

where $\boldsymbol{C}$ is the set of propositional logic constraints defined over variables $\boldsymbol{X}$, p_i is the predicted probability of X_i, $\boldsymbol{x} \models \boldsymbol{C}$ refers to the assignment $\boldsymbol{x}$ of variables $\boldsymbol{X}$ that satisfies constraints in $\boldsymbol{C}$, and $i : \boldsymbol{x} \models X_i$ (or $i : \boldsymbol{x} \models \neg X_i$)

refers to all the indices i where X_i is set to true (or false) in assignment $\boldsymbol{x}$.

As for our type constraints, the variables $\boldsymbol{X} \in \{0,1\}^{2R}$, where R is the number of relations, are defined over a pair of relations (r_1, r_2) which are assigned to 2 entity pairs ($subj_1$, obj_1) and ($subj_2$, obj_2) respectively. Specifically, $X_i \in \boldsymbol{X}_{1...R}$ equals 1 only when r_1 is the i^{th} relation, and $X_{R+i} \in \boldsymbol{X}_{R+1...2R}$ equals 1 only when r_2 is the i^{th} relation. We thereby define three semantic loss terms: $m^{ts}L^s(\boldsymbol{C}^{ts}, \boldsymbol{p})$, $m^{to}L^s(\boldsymbol{C}^{to}, \boldsymbol{p})$, $m^{tso}L^s(\boldsymbol{C}^{tso}, \boldsymbol{p})$, where m^{ts}, m^{to}, m^{tso} are 0-1 masks that are set to 1 when $subj_1 = subj_2$, $obj_1 = obj_2$, $subj_1 = obj_2$ respectively.

As for our cardinality constraints, the variables $\boldsymbol{X} \in \{0,1\}^{R}$ are defined over a pair of a relations r that is assigned to 2 entity pairs ($subj_1$, obj_1) and ($subj_2$, obj_2) simultaneously. Specifically, $X_i \in \boldsymbol{X}$ equals 1 only when r is the i^{th} relation, We thereby define two semantic loss terms: $m^{co}L^s(\boldsymbol{C}^{co}, \boldsymbol{p})$ and $m^{cs}L^s(\boldsymbol{C}^{cs}, \boldsymbol{p})$, where m^{co} and $m^{cs}$are 0-1 masks that are set to 1 when $subj_1 = subj_2 \wedge obj_1 \neq obj_2$ and $subj_1 \neq subj_2 \wedge obj_1 = obj_2$ respectively.

Note that, for each loss term, all the relation assignment pairs that satisfiest the corresponding constraints are included. Therefore, minimizing these semantic loss terms actually increases the likelihood of all the relation predictions that satisfies the type constraints.

TODO: state the commonality of the constraints in the front of the subsection. e.g., all defined on pairs of triples.

Training Procedure Here we introduce the learning and optimization details of our model. Our objective function is consists of two parts:

$$J(\theta) = J_{en}(\theta) + J_{SL}(\theta) \tag{2}$$

where $J_{en}(\theta)$ is the original cross-entropy classification loss, and $J_{SL}(\theta)$ is our semantic loss.

For any combination of two entity pair in a batch, we use their APCNN output to get the probability vector $\boldsymbol{p}$ in Eq. 1, and thereby compute the semantic loss terms $m^*L^s(\boldsymbol{C}^*, \boldsymbol{p})$. Here $m^*L^s(\boldsymbol{C}^*, \boldsymbol{p})$ denotes all the semantic loss terms defined in Sec. 2.3. Finally the semantic loss is defined as follows:

$$J_{SL}(\theta) = \sum_{i \neq j} m^*L^s(\boldsymbol{C}^*, \boldsymbol{p}_{ij}) \tag{3}$$

where $0 \leq i, j \leq batch\_size$, $\boldsymbol{p}_{ij}$ means the probability vector generated from the APCNN output of entity pair i and j.

During training, we iterate by randomly selecting a mini-batch from the training set until converge, and adopt the Adam optimizer (Kingma and Ba, 2014) to minimize the objective function.

Simplified Semantic Loss In our experiments, we find that the way (Xu et al., 2017) calculating semantic loss is time-consuming, so we simplify the semantic loss calculation without losing much performance, which greatly reduces the time overhead.

In (1), we need calculate semantic loss for every $\boldsymbol{x}(\boldsymbol{x} \models \alpha)$, but we find that for each entity pair combination, there exists $\boldsymbol{x}$ that is more important than others. So we use the gold information of entity pairs to find the most important positive rule, and random sample some rules as a supplement. This significantly reduces the computational complexity.

3 Experiments

3.1 Experimental Settings

Datasets We evaluate our approach on two datasets, including one English datasets and one Chinese dataset.

The English dataset, DBpedia dataset, is the one generated by (Chen et al., 2014), by mapping the triples in DBpedia (Bizer et al., 2009) to the sentences in New York Time corpus. It has 51 different relations, includes about 50,000 entity tuples, 134,000 sentences for training and 30,000 entity tuples, 53,000 sentences for testing.

The Chinese dataset, HudongBaiKe dataset, is also generated by (Chen et al., 2014), they derive knowledge facts and construct a Chinese KB from the Infoboxes of HudongBaiKe, one of the largest Chinese online encyclopedias. They collect four national economic newspapers in 2009 as their corpus. 28 different relations are mapped to the corpus and this results in 60,000 entity tuples, 120,000 sentences for training and 40,000 tuples, 83,000 sentences for testing.

Hyperparameters We use a grid search to determine the optimal parameters. For the Base Model(CNN), the CNN window size is 3, the Sentence embedding size is 256, position dimension is 5, batch size is 50, and the word embedding size is 50 for DBpedia dataset, 300 for Chinese dataset.

3.2 Experimental Results

We use the Precision-Recall curve as the evaluation criterion in our experiments. To prove our approach make sense, we compare our results with baseline CNN model (Lin et al., 2016) and ILP method (Chen et al., 2014). The results on DBpedia Dataset and Chinese Dataset are showed in Figure 1 and Figure 2.

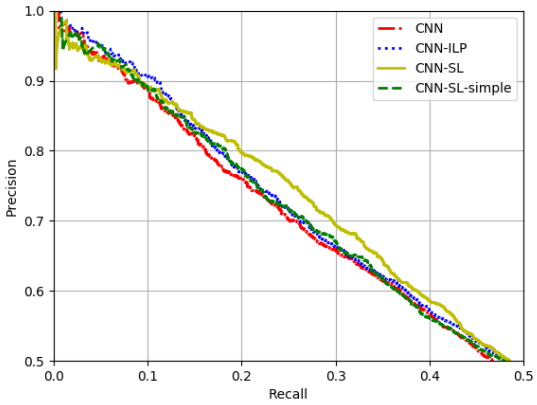

Figure 1: The DBpedia Dataset

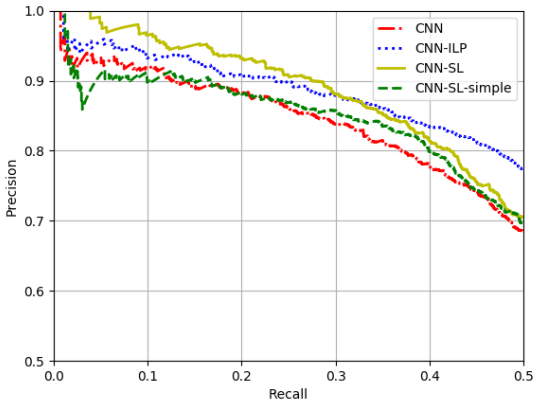

Figure 2: The Chinese Dataset

Compare with Base Model Figure 1 shows that compared with the baseline, our framework performs consistently better in the DBpedia dataset and Chinese dataset. In order to father demonstrate the semantic loss term helps to encode the constraints into relation extraction model, we count the tuple pairs that violates relation constraints introduced in Sec. 2.2, the results is show in Table 1.

Compare with ILP Method Compare the semantic loss method and ILP.

NOTE: clw also use cardinality constraints

Table 1: violate count

	Model	Type Violates	Cardinality Violates	Total
DBpedia	CNN	-	-	-
	CNN-SL	-	-	-
Chinese	CNN	-	-	-
	CNN-SL	-	-	-

Compare with Simplified Semantic Loss Compare the semantic loss method and the simplified semantic loss. Both on PR-curve, and on time.

4 Conclusion

Describe contribution, experimental results (effectiveness of semantic loss method), possibly future work.

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Liwei Chen, Yansong Feng, Songfang Huang, Yong Qin, and Dongyan Zhao. 2014. Encoding relation requirements for relation extraction via joint inference. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 818–827.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*, volume 1, pages 2124–2133.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Fabian Suchanek, James Fan, Raphael Hoffmann, Sebastian Riedel, and Partha Pratim Talukdar. 2013. Advances in automated knowledge base construction. *SIGMOD Records journal, March*.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. 2017. A semantic loss function for deep learning with symbolic knowledge. *arXiv preprint arXiv:1711.11157*.

Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. 2017. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of the*

55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1810–1820. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, pages 1753–1762.